Camera calibration
Triangulation
Perspective projection in homogenous coordinates

$$\vec{x}_{img} \equiv [I \ 0] \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \vec{x}_w$$

$$\vec{x}_{img} \equiv [R \ t] \vec{x}_w$$
Matrix transformations in 2D

\[ \vec{x}_{img} \equiv K \begin{bmatrix} R & t \end{bmatrix} \vec{x}_w \]

\[ K = \begin{bmatrix} 1 & 0 & t_u \\ 0 & 1 & t_v \\ 0 & 0 & 1 \end{bmatrix} \]

Translation

Scaling of Image x and y (conversion from “meters” to “pixels”)

\[ K = \begin{bmatrix} s_x & 0 & t_u \\ 0 & s_y & t_v \\ 0 & 0 & 1 \end{bmatrix} \]

Added skew if image x and y axes are not perpendicular
Final perspective projection

Camera extrinsics: where your camera is relative to the world. Changes if you move the camera.

\[ \vec{x}_{img} \equiv K \begin{bmatrix} R & t \end{bmatrix} \vec{x}_w \]

Camera intrinsics: how your camera handles pixel. Changes if you change your camera.

\[ \vec{x}_{img} \equiv P \vec{x}_w \]
Final perspective projection

\[ \vec{x}_{img} \equiv K [ R \ t ] \vec{x}_w \]

Camera parameters

\[ \vec{x}_{img} \equiv P\vec{x}_w \]
Camera calibration

• Goal: find the parameters of the camera

\[ \vec{x}_{img} \equiv K \begin{bmatrix} R & t \end{bmatrix} \vec{x}_w \]

• Why?
  • Tells you where the camera is relative to the world/particular objects
  • Equivalently, tells you where objects are relative to the camera
  • Can allow you to "render" new objects into the scene
Camera calibration
Camera calibration

\[ \vec{x}_{\text{img}} \equiv P \vec{x}_w \]

- Suppose we know that (X,Y,Z) in the world projects to (x,y) in the image.
- How many equations does this provide?

\[
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix} \equiv P \begin{bmatrix}
  X \\
  Y \\
  Z \\
  1
\end{bmatrix}
\]

Need to convert equivalence into equality.
Camera calibration

\[ \vec{x}_{img} \equiv P \vec{x}_w \]

• Suppose we know that \((X, Y, Z)\) in the world projects to \((x, y)\) in the image.

• How many equations does this provide?

\[
\begin{bmatrix}
\lambda x \\
\lambda y \\
\lambda \\
\end{bmatrix}
= P
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]

Note: \(\lambda\) is unknown
Camera calibration

\[ \mathbf{x}_{img} \equiv P \mathbf{x}_w \]

- Suppose we know that \((X,Y,Z)\) in the world projects to \((x,y)\) in the image.
- How many equations does this provide?

\[
\begin{bmatrix}
\lambda x \\
\lambda y \\
\lambda
\end{bmatrix}
= 
\begin{bmatrix}
P_{11} & P_{12} & P_{13} & P_{14} \\
P_{21} & P_{22} & P_{23} & P_{24} \\
P_{31} & P_{32} & P_{33} & P_{34}
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]
Camera calibration

\[\mathbf{x}_{img} \equiv P\mathbf{x}_w\]

• Suppose we know that \((X, Y, Z)\) in the world projects to \((x, y)\) in the image.

• How many equations does this provide?

\[
\lambda x = P_{11}X + P_{12}Y + P_{13}Z + P_{14} \\
\lambda y = P_{21}X + P_{22}Y + P_{23}Z + P_{24} \\
\lambda = P_{31}X + P_{32}Y + P_{33}Z + P_{34}
\]
Camera calibration

\[ \vec{x}_{img} \equiv P\vec{x}_w \]

• Suppose we know that \((X,Y,Z)\) in the world projects to \((x,y)\) in the image.

• How many equations does this provide?

\[
(P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = P_{11}X + P_{12}Y + P_{13}Z + P_{14} \\
(P_{31}X + P_{32}Y + P_{33}Z + P_{34})y = P_{21}X + P_{22}Y + P_{23}Z + P_{24}
\]

• 2 equations!

• Are the equations linear in the parameters?

• How many equations do we need?
Camera calibration

\[(P_{31}X + P_{32}Y + P_{33}Z + P_{34})x = P_{11}X + P_{12}Y + P_{13}Z + P_{14}\]

\[XxP_{31} + YxP_{32} + ZxP_{33} + xP_{34} - XP_{11} -YP_{12} - ZP_{13} - P_{14} = 0\]

- In matrix vector form: \(Ap = 0\)
- 6 points give 12 equations, 12 variables to solve for
- But can only solve upto scale
Camera calibration

• In matrix vector form: $A\mathbf{p} = 0$
• We want non-trivial solutions
• If $\mathbf{p}$ is a solution, $\alpha \mathbf{p}$ is a solution too
• Let’s just search for a solution with unit norm

$$A\mathbf{p} = 0$$

s.t

$$\|\mathbf{p}\| = 1$$

• How do you solve this?
Camera calibration

• In matrix vector form: \( Ap = 0 \)
• We want non-trivial solutions
• If \( p \) is a solution, \( \alpha p \) is a solution too
• Let’s just search for a solution with unit norm

\[
Ap = 0 \\
\text{s.t.} \\
\|p\| = 1
\]
Camera calibration

• What happens if there are more than 6 points?
• What if there is noise in the point locations?

\[ \min_{p} \|Ap\|^2 \]

\[ \text{s.t.} \quad \|p\| = 1 \]
Camera calibration

• What happens if there are more than 6 points?
• What if there is noise in the point locations?

\[ \min_{p} p^T A^T A p \]
\[ \text{s.t.} \quad A p = 0 \]
\[ \|p\| = 1 \]

• Look at eigenvector of \( A^T A \) with the smallest eigenvalue!
Camera calibration

• >=6 points with known 3D coordinates + known image coordinates

\[ \begin{align*}
XXP_{31} + YxP_{32} + ZxP_{33} + xP_{34} - XP_{11} - YP_{12} - ZP_{13} - P_{14} &= 0
\end{align*} \]

• In matrix vector form: want \( Ap = 0 \)

• Resilience to noise:

\[
\min_{\mathbf{p}} \mathbf{p}^T A^T A \mathbf{p} \\
\text{s.t.} \\
\| \mathbf{p} \| = 1
\]

• Look at eigenvector of \( A^T A \) with the smallest eigenvalue!
Camera calibration

• We need 6 world points for which we know image locations

• Would any 6 points work?
  • What if all 6 points are the same?

• Need at least 6 non-coplanar points!
Camera calibration
Camera calibration

\[
\vec{x}_{img} \equiv P \vec{x}_w
\]

\[
\vec{x}_{img} \equiv K [R \ t] \vec{x}_w
\]

• How do we get K, R and t from P?
• Need to make some assumptions about K
• What if K is identity?
Camera calibration

\[ \vec{x}_{img} \equiv P \vec{x}_w \]

\[ \vec{x}_{img} \equiv K \begin{bmatrix} R & t \end{bmatrix} \vec{x}_w \]

• How do we get K, R and t from P?
• Need to make some assumptions about K
• What if K is upper triangular?

\[ K = \begin{bmatrix} s_x & \alpha & t_u \\ 0 & s_y & t_v \\ 0 & 0 & 1 \end{bmatrix} \]

Added skew if image x and y axes are not perpendicular
Camera calibration

• How do we get K, R and t from P?
• Need to make some assumptions about K
• What if K is upper triangular?

\[ K = \begin{bmatrix} s_x & \alpha & t_u \\ 0 & s_y & t_v \\ 0 & 0 & 1 \end{bmatrix} \]

• P = K [ R  t]
• First 3 x 3 matrix of P is KR
• “RQ” decomposition: decomposes an n x n matrix into product of upper triangular and rotation matrix
Camera calibration

• How do we get K, R and t from P?
• Need to make some assumptions about K
• What if K is upper triangular?
• $P = K \begin{bmatrix} R & t \end{bmatrix}$
• First 3 x 3 matrix of P is KR
• “RQ” decomposition: decomposes an n x n matrix into product of upper triangular and rotation matrix
• $t = K^{-1}P[:,2] \leftarrow$ last column of P
Camera calibration and pose estimation
Triangulation

• Suppose we have two cameras
  • Calibrated: parameters known
• And a pair of corresponding pixels
• Find 3D location of point!
Triangulation

• Suppose we have two cameras
  • Calibrated: parameters known
• And a pair of corresponding pixels
• Find 3D location of point!

\[ P^{(1)} \]

\[ (x_1, y_1) \]

\[ P^{(2)} \]

\[ (x_2, y_2) \]
Triangulation

\[
\begin{bmatrix}
  x_1 \\
  y_1 \\
  1
\end{bmatrix} \quad \xrightarrow{\vec{x}_{img}^{(1)} \equiv P^{(1)} \vec{x}_w} \quad \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\]

\[
\begin{bmatrix}
  x_2 \\
  y_2 \\
  1
\end{bmatrix} \quad \xrightarrow{\vec{x}_{img}^{(2)} \equiv P^{(2)} \vec{x}_w} \quad \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\]
Triangulation

\[ \vec{x}_{img}^{(1)} \equiv P^{(1)} \vec{x}_w \]

\[ \lambda x_1 = P_{11}^{(1)} X + P_{12}^{(1)} Y + P_{13}^{(1)} Z + P_{14}^{(1)} \]

\[ \lambda y_1 = P_{21}^{(1)} X + P_{22}^{(1)} Y + P_{23}^{(1)} Z + P_{24}^{(1)} \]

\[ \lambda = P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)} \]

\[ (P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)}) x_1 = P_{11}^{(1)} X + P_{12}^{(1)} Y + P_{13}^{(1)} Z + P_{14}^{(1)} \]

\[ X(P_{31}^{(1)} x_1 - P_{11}^{(1)}) + Y(P_{32}^{(1)} x_1 - P_{12}^{(1)}) + Z(P_{33}^{(1)} x_1 - P_{13}^{(1)}) + (P_{34}^{(1)} x_1 - P_{14}^{(1)}) = 0 \]
Triangulation

\[ \overrightarrow{x}_{img}^{(1)} \equiv P^{(1)} \overrightarrow{x}_w \]

\[ X(P^{(1)}_{31} x_1 - P^{(1)}_{11}) + Y(P^{(1)}_{32} x_1 - P^{(1)}_{12}) + Z(P^{(1)}_{33} x_1 - P^{(1)}_{13}) + (P^{(1)}_{34} x_1 - P^{(1)}_{14}) = 0 \]

\[ X(P^{(1)}_{31} y_1 - P^{(1)}_{21}) + Y(P^{(1)}_{32} y_1 - P^{(1)}_{22}) + Z(P^{(1)}_{33} y_1 - P^{(1)}_{23}) + (P^{(1)}_{34} y_1 - P^{(1)}_{24}) = 0 \]

- 1 image gives 2 equations
- Need 2 images!
- Solve linear equations to get 3D point location
Linear vs non-linear optimization

\[ \lambda x_1 = P_{11}^{(1)} X + P_{12}^{(1)} Y + P_{13}^{(1)} Z + P_{14}^{(1)} \]

\[ \lambda y_1 = P_{21}^{(1)} X + P_{22}^{(1)} Y + P_{23}^{(1)} Z + P_{24}^{(1)} \]

\[ \lambda = P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)} \]

\[ x_1 = \frac{P_{11}^{(1)} X + P_{12}^{(1)} Y + P_{13}^{(1)} Z + P_{14}^{(1)}}{P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)}} \]

\[ y_1 = \frac{P_{21}^{(1)} X + P_{22}^{(1)} Y + P_{23}^{(1)} Z + P_{24}^{(1)}}{P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)}} \]
Linear vs non-linear optimization

\[ x_1 = \frac{P_{11}^{(1)} X + P_{12}^{(1)} Y + P_{13}^{(1)} Z + P_{14}^{(1)}}{P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)}} \]

\[ y_1 = \frac{P_{21}^{(1)} X + P_{22}^{(1)} Y + P_{23}^{(1)} Z + P_{24}^{(1)}}{P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)}} \]

\[ (x_1 - \frac{P_{11}^{(1)} X + P_{12}^{(1)} Y + P_{13}^{(1)} Z + P_{14}^{(1)}}{P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)}})^2 + (y_1 - \frac{P_{21}^{(1)} X + P_{22}^{(1)} Y + P_{23}^{(1)} Z + P_{24}^{(1)}}{P_{31}^{(1)} X + P_{32}^{(1)} Y + P_{33}^{(1)} Z + P_{34}^{(1)}})^2 \]

Reprojection error
Linear vs non-linear optimization

Reprojection error is the squared error between the true image coordinates of a point and the projected coordinates of hypothesized 3D point

Actual error we care about

Minimize total sum of reprojection error across all images

Non-linear optimization